

Course Title: Intermediate Java Android Game Programming

Course Overview

This course is designed for intermediate developers who want to delve deeper into game development using Java on the Android platform. By the end of this course, you will have a solid understanding of game mechanics, graphics rendering, user input handling, and game state management. You will also create a simple 2D game as a final project.

Course Structure

The course is divided into four modules, each focusing on different aspects of game development.

Module 1: Introduction to Game Development in Android

****Duration:**** 1 Week

Topics Covered:

- Overview of Game Development
- Setting Up the Development Environment
- Introduction to Android Game Libraries (LibGDX, AndEngine)
- Basic Game Loop Structure

Code Example:

```
```java
public class GameView extends SurfaceView implements Runnable {
 private Thread gameThread;
 private boolean isPlaying;

 public GameView(Context context) {
 super(context);
 }

 @Override
 public void run() {
 while (isPlaying) {
 update();
 draw();
 control();
 }
 }
}
```

```

private void update() {
 // Update game logic
}

private void draw() {
 // Draw the game
}

private void control() {
 // Control the frame rate
}
}
...

```

---

### ### Module 2: Graphics and Animation

**\*\*Duration:\*\* 2 Weeks**

#### #### Topics Covered:

- Drawing Shapes and Images
- Using Canvas for Rendering
- Implementing Sprites and Animation
- Handling Frame Rate and Performance Optimization

#### #### Code Example:

```

```java
public class GameView extends SurfaceView {
    private Paint paint;
    private Bitmap playerBitmap;
    private int playerX, playerY;

    public GameView(Context context) {
        super(context);
        paint = new Paint();
        playerBitmap = BitmapFactory.decodeResource(getResources(), R.drawable.player);
        playerX = 100;
        playerY = 100;
    }

    @Override
    protected void onDraw(Canvas canvas) {

```

```

        super.onDraw(canvas);
        canvas.drawColor(Color.BLACK); // Clear the screen
        canvas.drawBitmap(playerBitmap, playerX, playerY, paint); // Draw player
    }
}
'''

```

Module 3: User Input and Game Mechanics

****Duration:** 2 Weeks**

Topics Covered:

- Handling Touch and Gesture Input
- Implementing Game Mechanics (Collision Detection, Scoring)
- Creating Game Objects and Managing States
- Sound Effects and Background Music

Code Example:

```

```java
@Override
public boolean onTouchEvent(MotionEvent event) {
 if (event.getAction() == MotionEvent.ACTION_DOWN) {
 // Handle touch event
 int touchX = (int) event.getX();
 int touchY = (int) event.getY();
 checkCollision(touchX, touchY);
 }
 return true;
}

private void checkCollision(int x, int y) {
 // Check if the touch coordinates collide with game objects
}
'''

```

---

### ### Module 4: Final Project - Building a Simple 2D Game

**\*\*Duration:\*\* 3 Weeks**

#### #### Topics Covered:

- Game Design Principles
- Integrating All Concepts Learned
- Testing and Debugging
- Publishing Your Game on Google Play Store

#### #### Project Outline:

- Create a simple 2D game (e.g., a platformer or a puzzle game)
- Implement levels, scoring, and game over mechanics
- Add sound effects and background music
- Prepare the game for release

#### #### Code Example (Game Over Logic):

```
```java
private void gameOver() {
    isPlaying = false;
    // Show game over screen
    // Optionally restart the game
}

private void restartGame() {
    // Reset game variables and states
    isPlaying = true;
    // Restart game loop
}
```
```

---

## ## Course Requirements

- Basic knowledge of Java programming
- Familiarity with Android development and Android Studio
- A passion for game development

## ## Course Materials

- Lecture slides and notes
- Sample code and project files
- Recommended reading materials and resources

## ## Assessment

- Weekly quizzes to test understanding
- A final project submission (a simple 2D game)
- Peer reviews and feedback sessions

## ## Conclusion

By completing this course, you will have a comprehensive understanding of intermediate-level game programming in Java for Android. You will be equipped with the skills to create your own games and publish them on the Google Play Store. Happy coding!